PAPER

Towards Comprehensive Support for Business Process Behavior Similarity Measure

Cong LIU[†], Qingtian ZENG[†a)], Hua DUAN[†b)], *Nonmembers*, Shangce GAO[††c)], *Member*, *and* Chanhong ZHOU[†], *Nonmember*

SUMMARY Business process similarity measure is required by many applications, such as business process query, improvement, redesign, and etc. Many process behavior similarity measures have been proposed in the past two decades. However, to the best of our knowledge, most existing work only focuses on the direct causality transition relations and totally neglect the concurrent and transitive transition relations that are proved to be equally important when measuring process behavior similarity. In this paper, we take the weakness of existing process behavior similarity measures as a starting point, and propose a comprehensive approach to measure the business process behavior similarity based on the so-called Extended Transition Relation set, ETR-set for short. Essentially, the ETR-set is an extended transition relation set containing direct causal transition relations, minimum concurrent transition relations and transitive causal transition relations. Based on the ETR-set, a novel process behavior similarity measure is defined. By constructing a concurrent reachability graph, our approach finds an effective technique to obtain the ETR-set. Finally, we evaluate our proposed approach in terms of its property analysis as well as conducting a group of control experiments.

key words: *business process, petri nets, behavior similarity, concurrent reachability graph, ETR-set*

1. Introduction

Similarity measure for business process models have been suggested for different purposes such as measuring compliance between reference and actual models, searching for related models in a process repository, or locating services that adhere to a specification given by a process model [1]. For example, the China CNR Corporation Limited is a newly regrouped company which has over 20 subsidiary companies [37]. Before its establishment, ERP systems of its subsidiary companies which contain over 200,000 process models are developed independently. Currently, how to integrate the involved processes is really a big challenge for the corporation stakeholders. Automatically similarity measure among these processes will surely do a lot of favor. Therefore, how to determine or measure the similarity of business process models is an important step towards realizing these kinds of enterprise level applications.

In other words, we need to decide (1) if the behavior of these models are same, or (2) how different they are in case of disaffinity (e.g. the similarity degree [9]). Till now, some approaches that follow these two ideas have been presented to measure process similarity. For the former, trace equivalence [3], bisimulation [4], and branching bisimulation [5] have been proposed to match this goal. However, these approaches (e.g., [3]–[5]) only obtain results like if these models are same or not, but they are not able to distinguish how different these models are. Then following the latter, several different approaches, such as [6]–[15], are introduced to quantify the similarity degree between two process models. To be more specific, majorities of these works (e.g., [6]–[8]) try to capture the structural similarity, e.g., on the basis of graph edit distance [6], [7], tree edit distance [8], etc. However, as indicated in [9], two processes may look quite similar by considering the activity labels and the process structure, but may behave quite differently in their behavior perspective. In this way, the behavior characteristics of a process model can give a more comprehensive description of real execution of process model than those structure-based approaches. For this purposes, behavior-oriented process similarity measures (e.g., [10]–[15]) have aroused thriving public attention in recent years. Unfortunately, these methods are limited by analyzing the language (or to say transition firing sequences) generated by the models using their reachability graphs or directly derived from their corresponding event running log. To the best of our knowledge, the language of a process model can only represent the sequential behavior, and thereby the concurrent behavior are totally lost, which will surely exert a negative effect on the similarity measure. To make up for these defects, we propose a novel behavior similarity measure with full consideration of the concurrent activity relations. By building a concurrent reachability graph of the original process model, the concurrent can be effectively found.

The rest of this paper is structured as follows. Section 2 reviews some related work. Section 3 introduces some notations that are used throughout the paper. In Sect. 4, the weaknesses of existing behavior similarity measures are first discussed, and then ETR-set based similarity measure is introduced. In Sect. 5, an effective approach to obtain ETR-set is presented by constructing a concurrent reachability graph. Section 6 evaluates our approach, and finally Sect. 7 draws concluding remarks.

Manuscript received April 9, 2018.
Manuscript revised October 2, 2018.
Manuscript publicized December 5, 2018.

[†]The authors are with Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao 266590, China.

[††]The author is with University of Toyama, Toyama-shi, 930–8555 Japan.

a) E-mail: qtzeng@163.com
b) E-mail: huaduan59@163.com
c) E-mail: gaosc@eng.u-toyama.ac.jp

DOI: 10.1587/transinf.2018EDP7127

2. Related Work

A comprehensive survey on similarity measure of process models has been accomplished by Becker and Laue in [1], therefore, there is no need to give a complete overview of the related work. Here, we mainly summarize the work related to the behavior similarity measure.

In [10], [11], Bae et al. proposed a quantitative method by introducing a graph-based distance measure, which is named "graph dependency", for measuring the similarity and dis-similarity among various workflow designs. In their dependency graph, an arc is connected between two activities if one is a direct predecessor of the other. This approach can only represent the direct dependency relation of a process model while other important process information, e.g. the concurrent relation of activities is lost. In addition, this measure does not distinguish different types of connectors (such as AND, OR, XOR, and etc.), which has limited its application a lot. To represent the process models whose behavioral features have infinite traces in a finite way, Wang et al. [12] proposed a principle transition sequences based measure, which is denoted as PTS-measure. It is realized by constructing the coverability tree of a Petri net model and this method requires traversing all state space, which may lead to the so-called state space explosion problem. Moreover, the concurrent activity information cannot be reflected in the sequential trace set. In [13], Weidlich et al. proposed a behavior profile method by capturing the dependencies of activities, including strict order relation, exclusiveness relation, interleaving order relation and co-occurrence relation. It is proved that these profiles can be obtained in a cubic time complexity for sound free-choice processes. However, these are critical restrictions for real-life applicability, i.e. this method is not capable of handling models with OR-splits or unsound processes. In [14], Dijkman et al. presented causality graph which is composed of an activity set and its corresponding look-back and look-ahead links. By considering the behavior as a set of traces, the causal footprint is obtained by adapting causality graph. Based on this notion, a plug-in in the ProM framework is implemented to calculate process behavior similarities. However, this method will generate quite a number of useless intermediate results which influences its efficiency a lot.

To measure the behavior similarity of a workflow net, a transition adjacency relation set, short for TAR set, is introduced by Zha et al. in [15]. They argue that computing the whole set of firing sequences requires a large number of processing resources and the TAR set can be regarded as the gene of the all possible firing sequences, and therefore, can be used to substitute the full firing sequences to describe the behavior of workflow net models. This substitution has the following benefits: (1) the TAR set is able to specify the essence of all firing sequences; (2) it can be generated with less time and less process resources than obtaining all firing sequences; and (3) it is a finite set for real-life process cases with finite activities. For these reasons, their similarity measure is laid on the basis of TAR set. The TAR-similarity can be treated as an improvement of methods proposed by Bae et al. [10], [11] and Wang et al. [12], as it is capable of handling various types of connectors. However, the TAR set only contains direct causal relations to describe the behavior of processes. The concurrent and transitive dependencies are not included in the TAR set, which lead to incomplete description of process behavior.

As concluded by Becker and Laue in [1], it should be really appealing to supplement concurrent relations and transitive causal relations to the TAR-set when measuring similarities. In this paper, we take the above-mentioned weaknesses of TAR-similarity measure as a starting point, and extend the transition relations with both concurrent transition relations and transitive causal transition relations. As the first complete work to consider different kinds of transition dependency relations for measuring process behavior similarity, we are sure that our work can fill a critical blank in the literature, and therefore, improve the state-of-art to a great extent.

3. Preliminaries

This work is grounded on Petri nets and WF-nets. Some of the basic terminologies and notations of Petri nets [16], [17], [38]–[45] and WF-net [18] are listed as follows.

3.1 Petri Nets and WF-Net

Definition 1: A *Petri net* is a 4-tuple $\Sigma = (P, T, F, M_0)$, where (1) P is a finite set of places and T is a finite set of transitions such that $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$; (2) $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs (flow relation); and (3) $M_0 : P \rightarrow \{0, 1, 2, 3, \ldots\}$ is the initial marking.

As discussed in [17], a Petri net is a net structure mainly with two kinds of elements. One kind of element is place, which is graphically represented by circles. The second kind of element of a Petri net is transition, which is graphically represented by rectangles. Places and transitions are connected to each other by directed arcs. Graphically, an arc is represented by an arrow. Arcs run from a place to a transition or vice versa, never between places or between transitions. The places from which an arc runs to a transition are called the input places of the transition and the places to which arcs run from a transition are called the output places of the transition. Its formal specification is: for all $x \in P \cup T$, the set ${}^{\bullet}x = \{y | y \in P \cup T \wedge (y, x) \in F\}$ is the pre-set of x, and $x^{\bullet} = \{y | y \in P \cup T \wedge (x, y) \in F\}$ is the post-set of x. p is marked by M if $M(p) > 0$. A transition $t \in T$ is enabled under M, if and only if $\forall p \in^{\bullet} t : M(p) > 0$, denoted as $M[t >$. If $M[t >$ holds, t may fire, resulting in a new marking M', denoted as $M[t > M'$, such that $M'(p) = M(p) - 1$ if $p \in^{\bullet} t \setminus t^{\bullet}$, $M'(p) = M(p) + 1$ if $p \in t^{\bullet} \setminus^{\bullet} t$, and otherwise $M'(p) = M(p)$. A marking M is reachable from M_0 if there is a transition firing sequence δ such that $M_0[\delta > M$. In the following, we use $R(M_0)$ to represent the set of all reachable markings from M_0.

Definition 2: A Petri net $\Sigma = (P, T, F, M_0)$ is *bounded* if there is an integer k holding that $\forall M \in R(M_0)$ where $R(M_0)$ represents the set of all reachable states from M_0, $\forall p \in P$ such that $M(p) \leq k$, we say Σ is bounded. Otherwise, Σ is unbounded. If $\forall p \in P$ such that $M(p) \leq 1$, then we say that Σ is *safe*.

Definition 3: A Petri net $\Sigma = (P, T, F, M_0)$ is a *WF-net* if: (1) there is one source place $i \in P$ such that ${}^{\bullet}i = \emptyset$; (2) there is one sink place $o \in P$ such that $o^{\bullet} = \emptyset$; (3) each node $x \in P \cup T$ is on a path from i to o; and (4) $\forall p \in P$, $M_0(p) = 1$ if $p = i$, and otherwise $M_0(p) = 0$.

In a WF-net, the transition set T is used to represent the activities in a process model, and the source place and sink place represent the start and end of the process respectively.

For the rest of this paper, we assume that the mentioned WF-net model is a kind of safe and initially marked Petri nets, and a WF-net is also referred to as a Petri net and denoted as $\Sigma = (P, T, F, M_0)$ without explicitly stating it. Moreover, as WF-net is used to model business processes, we synonymously use the term transition and activity.

3.2 Reachability Graph

The reachability graph is a useful technique for the property analysis of Petri nets. It is first proposed by Karp and Miller [19], and then Finkel [20] optimized the algorithm and obtained the minimal reachability graph. More recently, Ye et al. [21] present a novel approach to constructing a reachability graph. Next, we introduce the basic definitions and notations following [21].

Definition 4: A *graph* $G = (V, E)$ is composed of a non-empty set of vertices V and a set of edges E. If the edges are ordered pairs (v, w) of vertices (i.e. $v, w \in V$), then the graph is said to be directed; a directed graph G is finite if both V and E are finite.

Definition 5: Given a Petri net $\Sigma = (P, T, F, M_0)$, a *reachability graph* $RG(\Sigma)$ is a directed graph with $R(M_0)$ be the vertex set. $\forall M_i, M_j \in R(M_0)$, if $\exists t \in T$ such that $M_i[t > M_j$, then there is a directed edge from M_i to M_j in $RG(\Sigma)$, and the edge is labeled with t.

Although a Petri net is finite, the set of its reachable markings is not always finite. For instance, when a Petri net is unbounded, its number of tokens can be infinite, thus the set of reachable markings are infinite. In this paper, we focus on the bounded (actually safe) Petri net models. With this assumption, the set of all reachable markings is a finite set. In this case, we can use the reachability graph to describe the state changes of a process model.

4. ETR-Set Based Process Behavior Similarity Measure

In this section, we start from discussing the inappropriateness of the TAR-measure using some illustrating example processes, and then process behavior similarity measure on the basis of ETR-set is introduced.

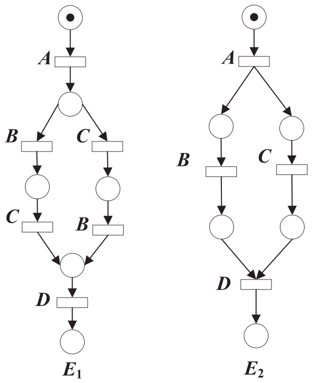


Fig. 1 Example Processes E_1 and E_2

4.1 Inappropriateness of the TAR-measure

Here, before rendering the weaknesses of the TAR measure, we will recall some basic definitions of TAR set and TAR-similarity following [15].

Definition 6: Let *FFS* be the full firing sequences of a WF-net $\Sigma = (P, T, F, M_0)$. Let $a, b \in T :< a, b >$ is a transition adjacency relation of Σ if there is a trace $\sigma = t_1 t_2 \dots t_n$ and $i \in 1, 2, \dots n$ such that $\sigma \in FFS$, $t_i = a$ and $t_{i+1} = b$. The complete TARs in *FFS* are called the TAR set.

Definition 7: Let $\Sigma_1 = (P_1, T_1, F_1, M_{01})$ and $\Sigma_2 = (P_2, T_2, F_2, M_{02})$ be two WF-nets and TAR_1 and TAR_2 be their TAR sets. The similarity between $\mathit{\Sigma_1}$ and $\mathit{\Sigma_2}$ is defined as $\mathit{Similarity}^T(\mathit{\Sigma_1}, \mathit{\Sigma_2}) = |\mathit{TAR_1} \cap \mathit{TAR_2}|/|\mathit{TAR_1} \cup \mathit{TAR_2}|$.

According to Definitions 6-7, it is easy to see that the TAR set involves (1) the direct causal transition relation; and (2) transition relations that happen to follow each other in some firing sequences, but they are not causally related in the model, e.g., the interleaving scenario. Therefore, this approach is not efficient enough to distinguish choice structures from concurrent structures. To illustrate this point in a more convincing way, we use the process models in [15] as shown in Fig. 1 as our motivating examples.

According to Definition 6, we can obtain the TAR set of E_1 and E_2 as $TAR_1 = \{AB, AC, BC, CB, BD, CD\}$ and $TAR_2 = \{AB, AC, BC, CB, BD, CD\}$. Then following Definition 7, the similarity between E_1 and E_2 is $Similarity^T(E_1, E_2) = |TAR_1 \cap TAR_2|/|TAR_1 \cup TAR_2| = 1$. This result shows that E_1 and E_2 are behaviorally similar. However, is that the case? Intuitively, transitions B and C in E_1 are in different causal relations which are determined by different routing conditions while transitions B and C in E_2 are in a concurrent relation for any situation. In this way, we argue that this method results in an inappropriate similarity measure. As far as we know, this is because the TAR set only contains the direct causal relations among transitions and the concurrent transition relations are totally neglected. Obviously, concurrent relation is

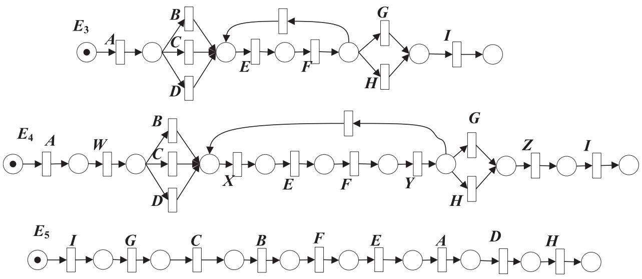


Fig. 2 Example Processes E_3, E_4 and E_5

of vital importance and indispensable in describing the behavior of process models. Next, we consider another group of example processes which comes from [1] as shown in Fig. 2. Similarly, the TAR sets of E_3, E_4 and E_5 are $TAR_3 = \{AB, AC, AD, BE, CE, DE, EF, FE, FG, FH, GI, GH\}$, $TAR_4 = \{AW, WB, WC, WD, BX, CX, DX, XE, EF, FY, YG, YH, YX, GZ, HZ, ZI\}$, and $TAR_5 = \{IG, GC, CB, BF, FE, EA, AD, DH\}$. Then, the similarity between E_3 and E_4 is $Similarity^T(E_3, E_4) = |TAR_3 \cap TAR_4|/|TAR_3 \cup TAR_4| = 0.04$ and the similarity of between E_3 and E_5 is $Similarity^T(E_3, E_5) = |TAR_3 \cap TAR_5|/|TAR_3 \cup TAR_5| = 0.11$.

We have $Similarity^T(E_3, E_4) < Similarity^T(E_3, E_5)$, which is not our expected results. Obviously, E_3 and E_4 are more similar in terms of behavior than that of E_3 and E_5. The reason lies on the fact that TAR set only contains direct causal information, but the indirect (or transitive) dependencies are overlooked. It will surely do much help by incorporating transitive causal relations into the computation of process behavior similarity. This means that rather than analyzing information like *transition t_i can be succeeded directly by transition t_j*, we also consider factors such as *after executing transition t_i, t_j will fire later*.

4.2 ETR-Set Based Process Behavior Similarity Measure

Because of missing the concurrent and transitive information of transitions when describing the process behavior, TAR-measure may cause some inaccuracies, or even unreasonable results. Therefore, it is necessary to find a complementary solution. To achieve this goal, we present a novel process behavior similarity measure based on an extended transition relation set, denoted as ETR-set. It involves three subsets: a direct causal transition relation set, a minimum concurrent transition relation set, and a transitive causal transition relation set.

Definition 8: Let $\Sigma = (P, T, F, M_0)$ be a WF-net and $\forall t_i, t_j \in T$, t_i and t_j are in a direct causal transition relation, also denoted as (t_i, t_j), if transition t_i is followed directly by transition t_j in Σ. All direct causal transition relations in Σ forms the *direct causal transition relation set*, which is denoted as *DCTR-set* $= \{(t_i, t_j) | t_i \textit{ is directly followed by } t_j\}$.

Definition 9: Let $\Sigma = (P, T, F, M_0)$ be a WF-net and $\forall t_i, t_j \in T$, t_i and t_j are in a transitive causal transition relation, denoted as (t_i, t_j), if transition t_i can be indirectly followed by transition t_j in Σ. All transitive causal transition relations in Σ form the *transitive causal transition relation set*, which is denoted as *TCTR-set* $= \{(t_i, t_j) | t_i \textit{ is indirectly followed by } t_j\}$.

According to Definitions 8-9, the *DCTR-set* and *TCTR-set* of E_3 are *DCTR-set*$_3$ $= \{(A, B), (A, C), (A, D), (B, E), (C, E), (D, E), (E, F), (F, E), (F, G), (F, H), (G, I), (H, I)\}$ and *TCTR-set*$_3$ $= \{(A, E), (A, F), (A, G), (A, H), (A, I), (B, F), (B, G), (B, H), (B, I), (C, F), (C, G), (C, H), (C, I), (D, F), (D, G), (D, H), (D, I), (E, G), (E, H), (E, I), (F, I)\}$.

Definition 10: Let $\Sigma = (P, T, F, M_0)$ be a WF-net, and $R(M_0)$ be the set of reachable markings. $\forall M \in R(M_0)$, $\forall t_i, t_j \in T$, t_i and t_j are in concurrent relation at marking M, denoted as $t_i//t_j$, if: (1) $M[t_i >$ and $M[t_j >$; and (2) $M[t_i > M_i \rightarrow M_i[t_j >$ and $M[t_j > M_j \rightarrow M_j[t_i >$.

Take E_2 in Fig. 1 as an example, its initial marking is $M_0 = (1, 0, 0, 0, 0, 0)$. According to Definition 10, we have $B//C$ at marking M_1 which is obtained by $M_0[A > M_1$ and $M_1 = (0, 1, 1, 0, 0, 0)$. Then, we extend this definition to an arbitrary number of transitions.

Definition 11: Let $\Sigma = (P, T, F, M_0)$ be a WF-net, and $R(M_0)$ be the set of reachable markings. $\forall M \in R(M_0)$, $\forall t_1, t_2 \ldots t_k \in T$, t_1, $t_2 \ldots t_k$ are in concurrent relation at marking M, denoted as $t_1//t_2 \ldots //t_k$, if: (1) $M[t_1 >, M[t_2 > \ldots M[t_k >$; and (2) $\forall t_i \in T$, let $M[t_i > M_i$, t_1, $t_2, \ldots$, t_{i-1}, $t_{i+1}, \ldots$, t_k are concurrent at M_i and let marking M_{i+1} be the firing result of t_1, $t_2, \ldots$, t_{i-1}, $t_{i+1}, \ldots$, t_k at marking M, $M_{i+1}[t_i >$.

Definition 12: Let $\Sigma = (P, T, F, M_0)$ be a WF-net and $\forall t_i, t_j \in T$, t_i and t_j are in a *minimum concurrent transition relation*, denoted as $t_i//t_j$, if transitions t_i and t_j can execute concurrently in Σ. All minimum concurrent transition relations in Σ form the minimum concurrent transition relation set, which is denoted as *MCTR-set* $= \{(t_i//t_j) | t_i \textit{ and } t_j \textit{ are in a concurrent transition relation}\}$.

As mentioned in Definitions 11, the concurrency may exist among more than two transitions. For example, when $t_i//t_j//t_k$ are in concurrency, then following Definition 12, we have *MCTR-set* $= \{(t_i//t_j), (t_j//t_k), (t_i//t_k)\}$.

Definition 13: Let $\Sigma = (P, T, F, M_0)$ be a WF-net, *ETR-set* = *DCTR-set*$\cup$*TCTR-set*$\cup$*MCTR-set* is named as its *extended transition relation set*, such that *DCTR-set* represents its direct causal transition relation set, *TCTR-set* represents its transitive causal transition relation set and *MCTR-set* represents its minimum concurrent transition relation set.

Definition 14: Let $\Sigma_1 = (P_1, T_1, F_1, M_{01})$ and $\Sigma_2 = (P_2, T_2, F_2, M_{02})$ be two WF-nets and *ETR-set*$_1$ and *ETR-set*$_2$ be their extended transition relation sets. Then the behavior similarity between Σ_1 and Σ_2 is defined as: $Sim(\Sigma_1, \Sigma_2) = |ETR\text{-}set_1 \cap ETR\text{-}set_2| / |ETR\text{-}set_1 \cup ETR\text{-}set_2|$.

Definition 15: Let $\Sigma_1 = (P_1, T_1, F_1, M_{01})$ and $\Sigma_2 = (P_2, T_2, F_2, M_{02})$ be two WF-nets and *ETR-set*$_1$ and *ETR-set*$_2$ be their extended transition relation sets. Then the behavior distance between Σ_1 and Σ_2 is defined as: $Dis(\Sigma_1, \Sigma_2)$=1−$|ETR\text{-}set_1 \cap ETR\text{-}set_2|/|ETR\text{-}set_1 \cup ETR\text{-}set_2|$.

We argue that the use of ETR-set to measure process behavior similarity has at least the following advantages: (1) it describes the process behavior in a more precise manner, including direct causal transition relations, concurrent transition relations and transitive causal transition relations. This will surely lead to a more accurate and reasonable similarity measure result; and (2) we can easily and effectively obtain a finite ETR-set for any process model regardless of structure limitations.

5. An Efficient Approach to Generate the ETR-Set

In this section, we propose an effective approach to compute the ETR-set. As mentioned in [15], Zha et al. applied the reachability graph of a WF-net to derive the TAR set. Even though the reachability graph is a powerful tool to analyze Petri nets, one of the biggest limitations is that the concurrent relations in Petri nets cannot be represented by traditional reachability graph. In order to represent the concurrent relation of the Petri net, we extend the traditional reachability graph to contain concurrent information and propose the notion of *Concurrent Reachability Graph*, short for CRG. In the following, we first define some of the basic concepts of CRG and then propose our ERT-set generation algorithm.

Definition 16: Let $\Sigma = (P, T, F, M_0)$ be a WF-net, $\forall t_1, t_2 \ldots, t_k \in T$, $k \in 1, 2, \ldots, n$, such that $t_1 // t_2 \ldots // t_k$ at marking M, we name $\{t_1, t_2, \ldots, t_k\}$ as a *concurrent transition unit*, short for *CTU* satisfying $CTU \subseteq T$. $CT(\Sigma) = \{CTU_i | i \in 1, 2, 3 \ldots\}$ is the concurrent transition set of Σ.

Obviously, $\{B, C\}$ is a *CTU* of E_2 in Fig. 1.

Given WF-net $\Sigma = (P, T, F, M_0)$ with $|p| = m$, for any $t \in T$, we have (1) $M_{in}(t) =< I_1, I_2, \ldots, I_m >$ is the input vector of t, such that $I_i = 1$ if $p_i \in^{\bullet} t$, and otherwise $I_i = 0$; and (2) $M_{out}(t) =< O_1, O_2, \ldots, O_m >$ is the output vector of t, such that $O_i = 1$ if $p_i \in t^{\bullet}$, and otherwise $O_i = 0$.

Taking the E_2 in Fig. 1 as an example, the input and output vectors of all transitions are obtained as follows: $M_{in}(A) =< 1, 0, 0, 0, 0, 0 >$, $M_{in}(B) =< 0, 1, 0, 0, 0, 0 >$, $M_{in}(C) =< 0, 0, 1, 0, 0, 0 >$, $M_{in}(D) =< 0, 0, 0, 1, 1, 0 >$, $M_{out}(A) =< 0, 1, 1, 0, 0, 0 >$, $M_{out}(B) =< 0, 0, 0, 1, 0, 0 >$, $M_{out}(C) =< 0, 0, 0, 0, 1, 0 >$ and $M_{out}(D) =< 0, 0, 0, 0, 0, 1 >$.

Definition 17: Let $\Sigma = (P, T, F, M_0)$ be a WF-net, and $R(M_0)$ be the set of reachable markings. $\forall M \in R(M_0)$, $\forall t_1, t_2, \ldots, t_k \in T$, $t_1, t_2 \ldots t_k$ are k concurrent transitions at marking M. The marking M' is called a *concurrent reachable marking* of Σ if $M[(t_1 // t_2 \ldots // t_k) > M'$.

Definition 18: Let $\Sigma = (P, T, F, M_0)$ be a WF-net, and $R(M_0)$ be the set of reachable markings. The *concurrent reachable marking set*, denoted as $CR(M_0)$, can be obtained by firing all *CTU*s in $CT(M_0)$.

Theorem 1: Let $\Sigma = (P, T, F, M_0)$ be a WF-net, and $R(M_0)$ be the reachable markings set, and $CR(M_0)$ be the concurrent reachable marking set. We prove that the $CR(M_0)$ is a subset of $R(M_0)$.

***Proof*.** $\forall M_{ci} \in CR(M_0)$, we have $M_0[CTU_i > \ldots CTU_j > M_{ci}$. As $CTU_i \subseteq T$, and we assume that $CTU_i = \{t_j, \ldots, t_k\}$. Similarity, we have $CTU_j = \{t_m, \ldots, t_n\}$. So $M_0[CTU_i > \ldots CTU_j > M_{ci} \Rightarrow M_0[t_j, \ldots, t_k > \ldots t_m, \ldots, t_n > M_{ci}$. Thus we have $M_{ci} \in R(M_0)$, i.e. $CR(M_0) \subseteq R(M_0)$. Therefore, $CR(M_0)$ is a subset of $R(M_0)$. ■

Algorithm 1 Obtain the $CT(M_0)$, $CR(M_0)$ and $CRG(\Sigma)$.

```
Input: $\Sigma = (P, T, F, M_0)$;
1: $CT(M_0) \leftarrow \emptyset$, $R(M_0) \leftarrow M_0$, $CR(M_0) \leftarrow M_0$, $ES(M_0) \leftarrow \emptyset$,
   $CRG(\Sigma) \leftarrow \emptyset$;
2: For each $t_i \in T$ do
       Calculate $M_{in}(t_i)$ and $M_{out}(t_i)$;
   End do
3: For each $M_i \in R(M_0)$ do
       (3.1) $R(M_0) \leftarrow R(M_0) - \{M_0\}$, $AET(M_i) \leftarrow \emptyset$;
       (3.2) For each $t_i \in T$ do
                 If $M_{in}(t_i) \leq M_i$ then
                  $AET(M_i) \leftarrow AET(M_i) \cup \{t_i\}$;
       (3.3) If $AET(M_i) == \emptyset$ then
                 Goto Step3;
       (3.4) For each $t_i \in AET(M_i)$ do
                 $M_j \leftarrow M_i - M_{in}(t_i) + M_{out}(t_i)$;
                 $R(M_0) \leftarrow R(M_0) \cup \{M_j\}$;
                 $ES(M_0) \leftarrow ES(M_0) \cup \{(M_i, M_j), t_i\}$;
       (3.5) $PS \leftarrow \emptyset$;
             $PS \leftarrow PS \cup PowerSet(AET(M_i))$;
             For each $ps_i \in PS$ do
                If $|ps_i < 2|$ then
                   $PS \leftarrow PS - \{ps_i\}$;
       (3.6) For each $ps_i \in PS$ do
             $InputVector \leftarrow \mathbf{0}$, $OutputVector \leftarrow \mathbf{0}$;
             For each $t_i \in ps_i$ do
                $InputVector \leftarrow InputVector + M_{in}(t_i)$;
                $OutputVector \leftarrow +M_{our}(t_i)$;
              If $M_i \geq InputVector$ then
                $M_j \leftarrow M_i - InputVector + OutputVector$;
                $CT(M_0) \leftarrow CT(M_0) \cup ps_i$;
                $CR(M_0) \leftarrow CR(M_0) \cup \{M_j\}$;
                $ES(M_0) \leftarrow ES(M_0) \cup \{(M_i, M_j), ps_i\}$;
4: $CRG(\Sigma) \leftarrow \{R(M_0), ES(M_0)\}$;
5: return $CT(M_0)$, $CR(M_0)$ and $CRG(\Sigma)$.
```

Algorithm 1 aims to construct the concurrent transition set $CT(M_0)$, the concurrent reachability markings $CR(M_0)$ and the concurrent reachability graph $CRG(\Sigma)$ from a safe Petri net. In Step 3.5, we use the function *PowerSet()* to compute the power set of the enabled transitions whose complexity is $o(2^{|T|})$. Therefore, the complexity of Algorithm 1 is $o(|CR(M_0)| * 2^{|T|})$ where $|T|$ is the number of transitions and $|CR(M_0)|$ is the number of reachable markings. Based on the complexity analysis, we can see that main complexity to obtain the $CRG(\Sigma)$ totally lies on the scale of reachable states. Also considering E_2, by executing Algorithm 1, we can obtain the concurrent transition set $CT(M_0) = \{(B//C)\}$, and the concurrent reachable

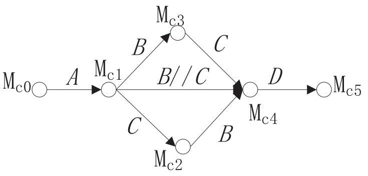


Fig. 3 Concurrent Reachability Graph of E_2

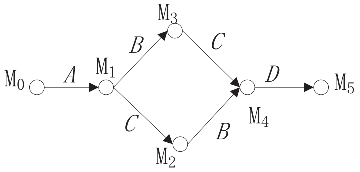


Fig. 4 Reachability Graph of E_2

marking set $CR(M_0) = \{M_{c4}\}$, and the concurrent reachability graph $CRG(E_2)$ as shown in Fig. 3. To give an intuitive comparison, the traditional reachability graph of E_2 is illustrated in Fig. 4. According to Figs. 3-4, we have $M_{c0} = M_0$, $M_{c1} = M_1$, $M_{c2} = M_2$, $M_{c3} = M_3$, $M_{c4} = M_4$ and $M_{c5} = M_5$. Also, $R(M_0) = \{M_0, M_1, M_2, M_3, M_4, M_5\}$, therefore we have $CR(M_0) \subseteq R(M_0)$, which is consistent with the conclusion of Theorem 1.

In the following, Algorithm 2 is presented to obtain the extended transition relation set. In Step 6, we use the *Warshell* Algorithm [23] to compute the transitive closure of the DCTR-set, whose complexity is $o(|DCTR\text{-}set|^3)$ with $|DCTR\text{-}set|$ be the number of direct causal relations. As we have $|DCTR\text{-}set| \leq |T|^2$, therefore the complexity of Step 6 is $o(|T|^6)$. Because $|CTU_i| \leq |T|$, the complexity of Step 2 is $o(|T|^3)$. Similarly, we can obtain the complexity of Steps 3-5 are all $o(|CR(M_0)| \times |T|^2)$. In this way, the complexity of Algorithm 2 is $o(|CR(M_0)| \times |T|^2)$ where $|T|$ is the number of transitions and $|CR(M_0)|$ is the number of reachable markings. Based on the complexity analysis, we can see that the main cost to obtain ETR-set also lies on the scale of reachable states. As we have mentioned in Sect. 2, our work is grounded on safe Petri nets (WF-nets). Therefore, we argue that the reachable markings are countable and thereby our methods can be realized in an acceptable time cost.

Considering E_2 in Fig. 2, by executing Algorithm 2, we can obtain $ETR\text{-}set_2 = DCTR\text{-}set_2 \cup TCTR\text{-}set_2 \cup MCTR\text{-}set_2$, where $DCTR\text{-}set_2 = \{(A, B), (A, C), (B, D), (C, D)\}$, $TCTR\text{-}set_2 = \{(A, D)\}$ and $MCTR\text{-}set_2 = \{(B//C)\}$.

Theorem 2: The proposed approach to generate the extended transition relation set has $o(|CR(M_0)| \times 2^{|T|})$ complexity where $|T|$ is the number of transitions and $|CR(M_0)|$ represents the number of reachable markings.
Proof. The ETR-set is generated by Algorithms 1-2 whose complexity is $o(|CR(M_0)| \times 2^{|T|})$ and $o(|CR(M_0)| \times |T|^2)$ respectively, and therefore the ETR-set can be obtained with $o(|CR(M_0)| \times 2^{|T|})$ complexity. ■

Algorithm 2 Obtain the ETR-set.

```
Input: $CRG(\Sigma) = \{R(M_0), ES(M_0)\}$ and $CT(M_0)$;
1: $DCTR\text{-}set \leftarrow \emptyset$, $TCTR\text{-}set \leftarrow \emptyset$ and $MCTR\text{-}set \leftarrow \emptyset$;
2: For each $CTU_i \in CT(M_0)$ do
        For each $t_m, t_n \in CTU_i$ do
            $MCTR\text{-}set \leftarrow MCTR\text{-}set \cup \{t_m//t_n\}$;
3: For each $M_i \in R(M_0)$ do
    (3.1) $ITS(M_i) \leftarrow \emptyset$;
    (3.2) For each $ES_j \in ES(M_0)$ do
        If $(M_{i-1}, M_i) \in ES_j$ then
            $ITS(M_i) \leftarrow ITS(M_i) \cup CTU\{M_{i-1}\}$;
4: For each $M_i \in R(M_0)$ do
    (4.1) $OTS(M_i) \leftarrow \emptyset$;
    (4.2) For each $ES_j \in ES(M_0)$ do
        If $(M_i, M_{i+1}) \in ES_j$ then
            $OTS(M_i) \leftarrow OTS(M_i) \cup CTU\{M_i\}$;
5: For each $M_i \in CR(M_0)$ do
        For each $t_i \in ITS(M_i)$ do
            For each $t_j \in OTS(M_i)$ do
                $DCTR\text{-}set \leftarrow DCTR\text{-}set \cup \{(t_i, t_j)\}$;
6: $TCTR\text{-}set \leftarrow$ warshall($DCTR\text{-}set$);
7: $ETR\text{-}set \leftarrow DCTR\text{-}set \cup TCTR\text{-}set \cup MCTR\text{-}set$;
8: return $ETR\text{-}set$.
```

6. Evaluation of ETR-Set Based Similarity Measure

6.1 Property Analysis of the ETR-Set0based Similarity Measure

Becker and Laue have proposed a series of desirable properties for similarity or distance measures in [1]. In this subsection, we will evaluate our ETR-set based similarity measure to decide whether it adheres to these properties or not.

Let Ψ be a set of process models, a distance measure is defined as: $\forall \Psi_1, \Psi_2 \in \Psi, Dis(\Psi_1, \Psi_2) = 1 - |ETR\text{-}set_1 \cap ETR\text{-}set_2| / |ETR\text{-}set_1 \cup ETR\text{-}set_2|$, based on Definition 15.

Property 1: Non-negative property, i.e. $Dis(\Psi_1, \Psi_2) \geq 0$.
Proof. Because $ETR\text{-}set_1 \cap ETR\text{-}set_2 \subseteq ETR\text{-}set_1 \cup ETR\text{-}set_2$, we have $|ETR\text{-}set_1 \cap ETR\text{-}set_2| \leq |ETR\text{-}set_1 \cup ETR\text{-}set_2|$. In addition, we have $|ETR\text{-}set_1 \cap ETR\text{-}set_2| / |ETR\text{-}set_1 \cup ETR\text{-}set_2| \leq 1$. Therefore, $Dis(\Psi_1, \Psi_2) = 1 - |ETR\text{-}set_1 \cap ETR\text{-}set_2| / |ETR\text{-}set_1 \cup ETR\text{-}set_2| \geq 0$. ■

Property 2: Symmetry property, i.e. $Dis(\Psi_1, \Psi_2) = Dis(\Psi_2, \Psi_1)$.
Proof. Because $ETR\text{-}set_1 \cap ETR\text{-}set_2 = ETR\text{-}set_2 \cap ETR\text{-}set_1$ and $ETR\text{-}set_1 \cup ETR\text{-}set_2 = ETR\text{-}set_2 \cup ETR\text{-}set_1$, we have $Dis(\Psi_1, \Psi_2) = 1 - |ETR\text{-}set_1 \cap ETR\text{-}set_2| / |ETR\text{-}set_1 \cup ETR\text{-}set_2| = 1 - |ETR\text{-}set_2 \cap ETR\text{-}set_1| / |ETR\text{-}set_2 \cup ETR\text{-}set_1| = Dis(\Psi_2, \Psi_1)$. ■

Property 3: If $Dis(\Psi_1, \Psi_2) = 0$ then Ψ_1 and Ψ_2 are the same.

Property 4: If $Dis(\Psi_1, \Psi_2) = 0$ then Ψ_1 and Ψ_2 have the same set of traces.

Property 5: If $Dis(\Psi_1, \Psi_2) = 0$ then Ψ_1 and Ψ_2 have the same ETR-sets.

Property 3 indicates that the distance between two models is 0 if and only if they are the same model, which is first defined by Santini and Jain in [24]. However, this rule is so strict that cannot be applied in real-life applications. Then this rule is relaxed as Property 4 by Becker and Laue in [1], it says that the distance of two models is 0 if and only if they have the same set of traces. In our previous discussion, we have indicated that the concurrent information of a process is neglected in its trace set. Therefore, we present Property 5 to reflect the behavioral equivalence of two processes in a more comprehensive. It means that $Dis(\Psi_1, \Psi_2)$ is 0 if and only if both models have the same ETR-sets, i.e. they have similar behaviors.

Property 6: Triangle inequality property, i.e. $Dis(\Psi_1, \Psi_3) \leq Dis(\Psi_1, \Psi_2) + Dis(\Psi_2, \Psi_3)$.

As indicated by Lin in [25], the triangle inequality property is not essential for similarity measures. Thus, we will not examine our approach with Property 6.

Property 7: Distance measure considers both commonalities and differences.

According to our formal definition for distance measure, it is easy to see that we have taken both commonality and difference of both models into account.

Property 8: Distance measure considers similarity measure between activities.

Our work is based on the assumption that a mapping between corresponding activities has been built. Therefore, Property 8 is not in our scope.

Property 9: Distance measure is defined for arbitrary process models.

As mentioned before, our work is built on Petri net theory, which suits various types of process structures, such as sequence, choice, concurrency, loop, etc.

Property 10: Distance measure can be computed efficiently.

In Theorem 2, we have discussed the time complexity of our method, and then draw the conclusion that it is in an exponential time cost.

According to the survey in [1], no existing measure fulfills all these desirable properties at the same time. From the above-mentioned properties, we can see that our ETR-set based similarity measure adheres to all of them except Property 8. This is because this property involves mapping activities which is not in the scope of this paper. Therefore, we are sure that our approach can work very well to measure process behavior similarity.

6.2 Experimental Evaluation

6.2.1 Evaluation Based on Synthetic Process Models

Process similarity measures are proposed for various purposes, and hardly a measure can fulfill or suit all kind of applications. For example, the measures [2] that do not consider the routing structures are useful to find related models from a process repository. While when we need to discover services, the methods based on measuring process behaviors [10]–[15] are more reasonable. Generally speaking, the process behavior can be described in two different perspectives: computing the whole set of traces as well as exploiting dependency relations between activities. Our measure belongs to the later one, as the former measure requires much processing resources. More specifically, our similarity measure is actually an improvement of the TAR-measure by adding the concurrent transition relations and transitive causal transition relations to the TAR-measure. Therefore, in this experimental evaluation section, we only compare the results between these two measures. In the following, we first use the example processes in Figs. 1-2 to illustrate how our similarity measure improves the TAR-measure. In Fig. 1, the similarity of E_1 and E_2 using TAR-measure is $Similarity^T(E_1, E_2) = 1$ while using our ETR set measure we have $Similarity^T(E_1, E_2) = 0.875$. Obviously, our result is more convincing as transitions B and C behave differently in E_1 and E_2, i.e. conditional causality versus concurrency. In Fig. 2, the similarity of E_3 and E_4 is $Similarity^T(E_3, E_4) = 0.04$ and that of E_3 and E_5 is $Similarity^T(E_3, E_5) = 0.11$. It shows that $Similarity^T(E_3, E_4) < Similarity^T(E_3, E_5)$ which is out of our expectations. Then based on our ETR-set measure we have $Similarity^T(E_3, E_4) = 0.46$ and $Similarity^T(E_3, E_5) = 0.18$, and we have $Similarity^T(E_3, E_4) > Similarity^T(E_3, E_5)$. Obviously, our result, i.e. E_3 and E_4 are more similar in terms of behavior than that of E_3 and E_5, is more practical as we take the transitive dependencies into account.

6.2.2 Evaluation Based on Control Experiment

Inspired by Becker and Laue [1], we conduct a set of control experiments by providing a moderately sized model E_0 and its variants E_6-E_{12} as shown in Fig. 5. It is composed of a set of hospital registration process models of different organizations. E_0 is the official reference model, and E_6-E_{12} are their corresponding variants from some local hospitals. Then we need to measure whether these hospitals design their business processes following the reference model.

The similarity results of the *TAR-measure* and our ETR-set measure are shown in Fig. 6. To compare them effectively, we also provide the similarity measure as perceived by human experts and domain analysts, i.e., the so-called ground truth knowledge. To obtain reliable and unbiased ground truth knowledge, we performed an interview. The interview involves ten interviewees (four master students, four PhD students, and two assistant professors) that have extensive experiences on business process modelling. Each interviewee is asked to independently measure the similarity between the process variants and the reference process. Then, we take the average of the all similarities as the final ground truth knowledge.

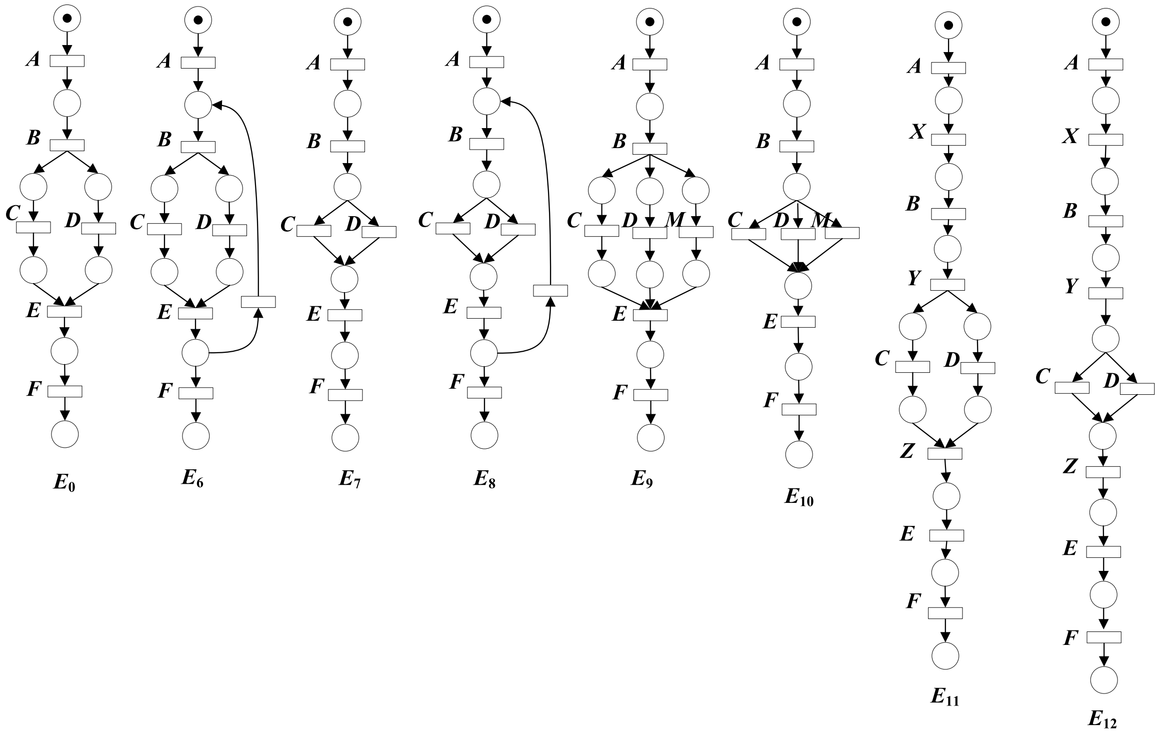


Fig. 5 Processes Models of E_0 and E_6-E_{12}

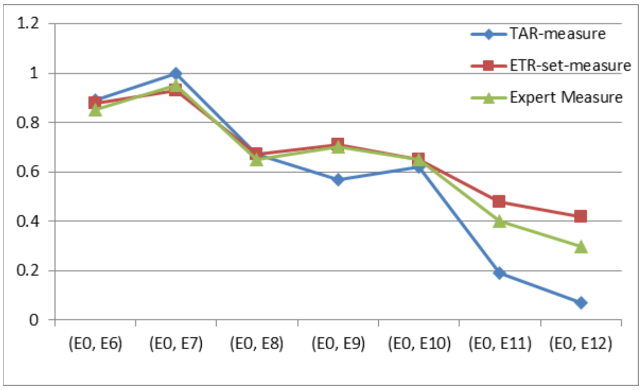


Fig. 6 Similarity Comparison between TAR-measure and ETR-set-measure

According to Fig. 6, we can see that the ETR-set-measure is much closer to the Expert Measure than TAR-measure, which means that the ETR-set-measure is more consistent with reality and expert perception. Moreover, the results based on both two measures are almost consistent for $Sim(E_0, E_6)$, $Sim(E_0, E_8)$ and $Sim(E_0, E_{10})$ while others such as $Sim(E_0, E_7)$, $Sim(E_0, E_9)$, $Sim(E_0, E_{11})$ and $Sim(E_0, E_{12})$ differ a lot. It reveals that the measure developed in the paper behaves more or less similar to the TAR measure. This makes sense as the measure developed in the paper is based on the TAR measure. In addition, our measure is in some way better than the TAR measure for some cases, e.g. models with current structures. It is easy to conclude that E_6, E_8 and E_{10} share the same behavior features, i.e. they do not contain concurrent activities. E_7, E_9, E_{11} and E_{12} also have some commonalities in their behavior, i.e., they either have concurrent activity relations (e.g., E_7 and E_9) or they are derived by adding some extra activities to the original model (e.g., E_{11} is obtained from E_0 by inserting activities X, Y and Z), which destroys their original direct dependency relations.

By experimental evaluation, we argue that our ETR-set similarity measure is more suitable to measure similarity between (1) processes with concurrent structures; and (2) processes that differ from each other by adding or deleting certain activities. A typical application of this scenario is to measure the similarity between cross-organizational processes that describe changeable business requirements at different stages.

7. Conclusion

This paper proposes a novel process behavior similarity measure based on ETR-set. As different process similarity measures have its specific application spectrum, our approach particularly suitable to compute similarity of process models that contain concurrent branches, and process models differ from each other by adding or deleting certain activities. The main contributions of this paper include:

- An extended transition relation set (ETR-set) is introduced to describe the behavior of a process. It can capture the direct causal transition relations, minimum concurrent transition relations and transitive causal transition relations, and then a novel technique to compute the similarity based on ETR-set is presented; and

- By building a concurrent reachability graph, corresponding algorithms are developed to find an effective technique to obtain the ETR-set.

Although this work are presented in the context of Petri nets (or to say WF-net), the approach can be applied to any process modeling language [33], e.g., [26], BPEL [27], EPCs [28], UML [29] and etc. This attributes to the fact that Petri net based formalizations can be expressed by most of the process modeling languages.

This work is based on the assumption that a mapping between corresponding activities has been built. Therefore, our similarity measure has not taken the similarity measure between activity nodes into consideration. In our future work, we aim to add the mapping techniques [14], [29]–[32] to improve our ETR-set based similarity measure.

It is the case that if we keep a mapping of the pairs of transitions with their corresponding type of behavior relation, then the ETR-set would become a sort of combination of the alpha-relations and the behavioral profiles [34]. In fact, DCTR-set and MCTR-set would correspond to causality and concurrency alpha relations whereas TCTR-set and MCTR-set would roughly correspond to strict causal and interleaving relations from behavioral profiles. As our work tries to incorporate the concurrent transition relation for business process similarity measure, other efficient approaches to obtain structue concurrency in Petri nets [36] are highly desired in the future. Additionally, experimental comparisons of our ETR-set-based similarity measure by tuning the weight of DCTR-set, TCTR-set and MCTR-set using real business processes is highly desired in the future.

Acknowledgments

This work was supported in part by NSFC (61472229, 61602279, 71704096 and 31671588), Science and Technology Development Fund of Shandong Province of China (2016ZDJS02A11, ZR2017BF015 and ZR2017MF027), the Taishan Scholar Climbing Program of Shandong Province, and SDUST Research Fund (2015TDJH102).

References

[1] M. Becker and R. Laue, "A comparative survey of business process similarity measures," Comput. Ind., vol.63, no.2, pp.148–167, 2012.

[2] Z. Yan, R. Dijkman, and P. Grefen, "Fast business process similarity search," Distrib. Parallel. Dat., vol.30, no.2, pp.105–144, 2012.

[3] L. Pomello, G. Rozenberg, and C. Simone, "A survey of equivalence notions for net based systems," Advances in Petri Nets 1992, Lecture Notes in Computer Science, vol.609, pp.410–472, Springer Berlin Heidelberg, 1992.

[4] R.J. Van Glabbeek and W.P. Weijland, "Branching time and abstraction in bisimulation semantics," J. ACM, vol.43, no.3, pp.555–600, 1996.

[5] U. Engberg and M. Nielsen, "A Calculus of Communicating Systems with Label Passing," DAIMI Report Series, vol.15, no.208, 1986.

[6] R. Dijkman, M. Dumas, and L. García-Bañuelos, "Graph Matching Algorithms for Business Process Model Similarity Search," Business Process Management, Lecture Notes in Computer Science, vol.5701, pp.48–63, 2009.

[7] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," Inform. Syst., vol.36, no.2, pp.498–516, 2011.

[8] J. Vanhatalo, H. Völzer, and J. Koehler, "The refined process structure tree," Data Knowl. Eng., vol.68, no.9, pp.793–818, 2009.

[9] W.M.P.V. Aalst, A.K.A. de Medeiros, and A.J.M.M. Weijters, "Process Equivalence: Comparing Two Process Models Based on Observed Behavior," Business Process Management, Lecture Notes in Computer Science, vol.4102, pp.129–144, Springer Berlin Heidelberg, 2006.

[10] J. Bae, L. Liu, J. Caverlee, and W.B. Rouse, "Process Mining, Discovery, and Integration using Distance Measures," 2006 IEEE International Conference on Web Services (ICWS'06), pp.479–488, 2006.

[11] J. Bae, L. Liu, J. Caverlee, L.-J. Zhang, and H. Bae, "Development of Distance Measures for Process Mining, Discovery and Integration," International Journal of Web Services Research, vol.4, no.4, pp.1–17, 2007.

[12] J. Wang, T. He, L. Wen, N. Wu, A.H.M.T. Hofstede, and J. Su, "A Behavioral Similarity Measure between Labeled Petri Nets Based on Principal Transition Sequences," On the Move to Meaningful Internet Systems: OTM 2010, Lecture Notes in Computer Science, vol.6426, pp.394–401, Springer Berlin Heidelberg, 2010.

[13] M. Weidlich, J. Mendling, and M. Weske, "Efficient Consistency Measurement Based on Behavioral Profiles of Process Models," IEEE Trans. Softw. Eng., vol.37, no.3, pp.410–429, 2011.

[14] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," Inform. Syst., vol.36, no.2, pp.498–516, 2011.

[15] H. Zha, J. Wang, L. Wen, C. Wang, and J. Sun, "A workflow net similarity measure based on transition adjacency relations," Comput. Ind., vol.61, no.5, pp.463–471, 2010.

[16] C. Liu, Q. Zeng, J. Zou, F. Lu, and Q. Wu, "Invariant Decomposition Conditions for Petri Nets Based on the Index of Transitions," Information Technology Journal, vol.11, no.7, pp.768–774, 2012.

[17] W. Reisig, "Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies," Springer, 2013.

[18] Q. Zeng, S.X. Sun, H. Duan, C. Liu, and H. Wang, "Cross-organizational collaborative workflow mining from a multi-source log," Decis. Support. Syst., vol.54, no.3, pp.1280–1301, 2013.

[19] R.M. Karp and R.E. Miller, "Parallel program schemata," J. Comput. Syst. Sci., vol.3, no.2, pp.147–195, 1969.

[20] A. Finkel, "The minimal coverability graph for Petri nets," Advances in Petri Nets 1993, Lecture Notes in Computer Science, vol.674, pp.210–243, Springer Berlin Heidelberg, 1993.

[21] X. Ye, J. Zhou, and X. Song, "On reachability graphs of Petri nets," Comput. Electr. Eng., vol.29, no.2, pp.263–272, 2003.

[22] T. Curran, G. Keller, and A. Ladd, "SAP R/3 business blueprint: understanding the business process reference model," Prentice-Hall, Inc., 1997.

[23] P. Purdom, "A transitive closure algorithm," Bit, vol.10, no.1, pp.76–94, 1970.

[24] S. Santini and R. Jain, "Similarity measures," IEEE Trans. Pattern Anal. Mach. Intell., vol.21, no.9, pp.871–883, 1999.

[25] D. Lin, "An information-theoretic definition of similarity," ICML, vol.98, pp.296–304, 1998.

[26] R.M. Dijkman, M. Dumas, and C. Ouyang, "Semantics and analysis of business process models in BPMN," Inform. Softw. Tech., vol.50, no.12, pp.1281–1294, 2008.

[27] N. Lohmann, "A feature-complete Petri net semantics for WS-BPEL 2.0," Web Services and Formal Methods, Springer Berlin Heidelberg, pp.77–91, 2008.

[28] J. Mendling, "Metrics for process models: Empirical foundations of verification, error prediction, and guidelines for correctness," Springer, 2008.

[29] R. Eshuis and R. Wieringa, "Tool support for verifying UML activ-

ity diagrams," IEEE Trans. Softw. Eng., vol.30, no.7, pp.437–447, 2004.

[30] M. Weidlich, R. Dijkman, and J. Mendling, "The ICoP Framework: Identification of Correspondences between Process Models," Active Flow and Combustion Control 2018, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, vol.141, pp.483–498, Springer Berlin Heidelberg, 2010.

[31] M. Ehrig, A. Koschmider, and A. Oberweis, "Measuring similarity between semantic business process models," Proceedings of the fourth Asia-Pacific conference on Comceptual modelling, vol.67, pp.71–80, 2007.

[32] M. Niemann, M. Siebenhaar, J. Eckert, and R. Steinmetz, "Process Model Analysis Using Related Cluster Pairs," Business Process Management Workshops, Lecture Notes in Business Information Processing, vol.66, pp.547–558, 2011.

[33] N. Lohmann, E. Verbeek, and R. Dijkman, "Petri Net Transformations for Business Processes – A Survey," Transactions on Petri Nets and Other Models of Concurrency II, Lecture Notes in Computer Science, vol.5460, pp.46–63, Springer Berlin Heidelberg, 2009.

[34] E. Badouel, "On the α-Reconstructibility of Workflow Nets," Petri Nets, vol.7347, pp.128–147, 2012.

[35] M. Weidlich and J.M.V. Werf, "On Profiles and Footprints – Relational Semantics for Petri Nets," Petri Nets, vol.7347, pp.148–167, 2012.

[36] A. Kovalyov and J. Esparza, "A polynomial algorithm to compute the concurrency relation of free-choice signal transition graphs," SFB, vol.342, 1995.

[37] China CNR Corporation Limited. http://www.chinacnr.com/LISTS/product/_MAINPAGE/englishmain/, 2008.

[38] C. Liu, Q. Zeng, H. Duan, M. Zhou, F. Lu, and J. Cheng, "E-Net Modeling and Analysis of Emergency Response Processes Constrained by Resources and Uncertain Durations," IEEE Trans. Syst. Man Cybern, Syst., Man, and Cybernetics: Systems, vol.45, no.1, pp.84–96, 2015.

[39] Q. Zeng, C. Liu, and H. Duan, "Resource conflict detection and removal strategy for nondeterministic emergency response processes using Petri nets," Enterprise Information Systems, vol.10, no.7, pp.729–750, 2016.

[40] C. Liu, J. Cheng, Y. Wang, and S. Gao, "Time Performance Optimization and Resource Conflicts Resolution for Multiple Project Management," IEICE Trans. Inf. & Syst., vol.E99-D, no.3, pp.650–660, 2016.

[41] C. Liu, H. Duan, Q. Zeng, M. Zhou, F. Lu, and J. Cheng, "Towards Comprehensive Support for Privacy Preservation Cross-organization Business Process Mining," IEEE Trans. Serv. Comput., pp.1–1, 2016. DOI: 10.1109/TSC.2016.2617331

[42] J. Cheng, C. Liu, M. Zhou, Q. Zeng, and A. Yla-Jaaski, "Automatic Composition of Semantic Web Services Based on Fuzzy Predicate Petri Nets," IEEE Trans. Automat. Sci. Eng., vol.12, no.2, pp.680–689, 2015.

[43] Q. Zeng, F. Lu, C. Liu, et al., "Modeling and verification for cross-department collaborative business processes using extended petri nets," IEEE Trans. Syst., Man, Cybern., Syst., vol.45, no.2, pp.349–362, 2015.

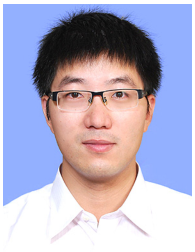
[44] C. Liu, Y. Pei, Q. Zeng, and H. Duan, "LogRank: An Approach to Sample Business Process Event Log for Efficient Discovery," W. Liu W, F. Giunchiglia, B. Yang (eds) Knowledge Science, Engineering and Management, KSEM 2018, Lecture Notes in Computer Science, vol.11061, pp.415–425, Springer, Cham.

[45] Q. Li, Y. Deng, C. Liu, Q. Zeng, and Y. Lu, "Modeling and analysis of subway fire emergency response: An empirical study," Safety. Sci., vol.84, pp.171–180, 2016.

Cong Liu received his M.S. degree in computer science and technology from Shandong University of Science and Technology, Qingdao, China in 2015. His research interests are in the areas of Business Process Management, Process mining and Petri nets.

Qingtian Zeng received the Ph.D. degree in computer software and theory from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with Shandong University of Science and Technology, Qingdao, China. His research interests are in the areas of Petri nets, process mining, and knowledge management.

Hua Duan is an associate professor at the College of Information Science and Technology, Shandong University of Science and Technology. She obtained her PhD in applied mathematics from Shanghai Jiaotong University in 2008. Her research interests are in the areas of Petri nets, Process Mining, and Machine Learning.



Shangce Gao received his Ph.D. degree in innovative life science from University of Toyama, Toyama, Japan in 2011. He is currently an Associate Professor with the Faculty of Engineering, University of Toyama, Japan. His main research interests include mobile computing, nature-inspired technologies, and neural networks for optimization and real-world applications. He was a recipient of the Shanghai Rising-Star Scientist award, the Chen-Guang Scholar of Shanghai award, the Outstanding Academic Performance Award of IEICE, and the Outstanding Academic Achievement Award of IPSJ. He is a Senior Member of IEEE.

ChangHong Zhou is currently pursuing the Ph.D. degree in computer software and theory from the Shandong University of Science and Technology, Qingdao, China. She is currently a Lecturer with the Shandong University of Science and Technology. Her current research interests include process recommendation and process similarity measure.